Remarks and Arguments

Claims 1-15 have been presented for examination. Claims 1, 2, 7, 11, 12, 14 and 15 have been amended. New claims 35-52 have been added.

Claims 1-4, 12, 14 and 15 have been rejected as obvious under 35 U.S.C. §103(a) over U.S. Patent No 5,806,075 (Jain) in view of U.S. Patent No. 6,449,622 (LaRue). The examiner comments that Jain discloses the claimed subject matter with the exception that Jain does not show receiving data change requests out of order. The examiner asserts that mechanisms for dealing with the reception of out-of-order data change requests is well-known in the art as evidenced by LaRue. Although LaRue relates to synchronizing datasets rather than peer-to-peer collaboration systems, the examiner claims that the Larue system relates to analogous art and discloses the coordination of data change requests that are received out of order. The examiner claims that a person skilled in the art would have recognized that incorporation of the features of LaRue into the Jain system would allow Jain to operate in high latency environments.

As previously discussed, in the present invention, a dynamics manager in each collaborator's computer implements a data change request priority scheme that determines an order for executing the data change requests to promote data consistency. If a data change request from a remote collaborator arrives at a local collaborator, the dynamics manager can determine (from the order and dependency information in the new request) that this newly-received data change request should have been made before other already-made data change requests. In this case, the dynamics manager effectively reorders the data change requests and makes the changes to the local data copy in the correct order. It does this by selectively undoing data changes that have already been made by "undoing" the effects of these changes change-by-change until the local data copy is in a state at which the newly received data change request should have been made. Each individual change can be undone because the dynamics manager saves the change and information to undo the change in a delta log.

After undoing the changes, the dynamics manager causes the newly received data change to be made and then remakes all of the subsequent data changes thereby resulting in a new order of data changes that incorporates the newly-received change. The result is that conflicts are resolved and the data changes are made in the correct order without user intervention.

The Jain reference discloses a procedure level replication technique that applies remote procedure calls (RPCs) to modify a replicated data copy and can address conflicts that occur during the processing of the RPCs and "rollback" any modifications made when a conflict is detected. In Jain, the processing of RPCs is "atomic", that is, the modification performed by the RPC is either made permanent or not performed. Jain discloses two methods for processing RPCs: transactional processing and non-transactional processing. Transactional processing processes a series of RPCs whereas non-transactional processing processes only a single RPC.

In both types of processing, a "savepoint" is established that saves the state of the data before one or more RPCs are applied. The savepoint is established for transactional processing as set forth in column 20, lines 7-13 and the savepoint for non-transactional processing is discussed at column 21, line 46. In transaction processing, if all RPC(s) complete without exception, then the transaction is "committed" and the changes made permanent as set forth at Jain column 20, lines 25-27 and at column 21, lines 58-60. Alternatively, if an exception occurs, then all of the transaction are rolled back to the savepoint as set forth in Jain, column 21, lines 11-13 and column 21, lines 49-50, and an entry is made in the exceptions table. After an exception occurs due to an update conflict, error handling can then be initiated. This error handling is described at Jain column 22, lines 47-55, which states that "The subsequent error processing can be done with various degrees of operator intervention and automation."

Jain discloses no mechanism for handling an RPC that might arrive "out-of-order" after RPCs have been committed because changes have been made permanent at that point. In particular, Jain discloses no mechanism for "undoing" changes on a change-by-change basis since Jain does not store undo information for each change and only stores the data state at the last savepoint.

The Larue reference supplements the Jain disclosure by disclosing that conflicts can be automatically handled by comparing timestamps on the conflicting changes when both are present during processing and implementing the change with the latest timestamp value, a "latest change wins" approach described at LaRue, column 47, lines 13-19. However, if this approach does not succeed because the timestamps cannot be compared, then Larue also uses user intervention as described at LaRue, column 47, lines 20-28. LaRue also does not address the situation where a new "out-of-order" change arrives after all other changes have been committed. Specifically, LaRue discloses no mechanism for "undoing" changes on a change-by-change basis since LaRue does not store undo information for each change and only stores the data state at the last savepoint.

Consequently, the combination of Jain and LaRue might suggest using a "latest change wins" approach to handling conflicts are an alternative to the rollback disclosed in Jain, but cannot teach or suggest undoing each change on a change-by-change basis since neither reference teaches or suggests this.

In order to prevent confusion regarding the "rollback" disclosed in Jain with the change-by-change undoing of the present invention, the claims have been amended to replace the term "rollback" with undo. For example, amended claim 1 recites, in lines 17-22, "...the making of selected data changes in an order, and, responsive to a data change request being received out of the order, the undoing of the selected data changes to a point where a data change corresponding to the out-of-order data change request should have been made and the remaking of the undone data changes in another order so that the local copy of the data is consistent with a copy of the data maintained by the remote network-capable device." Therefore, amended claim 1 clearly recites that data changes are undone back to the point where an out-of-order change should have been made and then the changes are redone to make the changes in a new order. As discussed above, the combination of Jain and LaRue cannot perform such an operation and does not suggest such an operation. Thus, amended claim 1 clearly distinguishes over the combination of the cited references.

Similar changes have been made in independent claims 12, 14 and 15 and these claims distinguish over the cited references in the same manner as amended claim 1.

Amendments have been made to claims 2, 7 and 11 to conform them to the changes made in claim 1.

Claims 2-4 are dependent, either directly or indirectly, on amended claim 1 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination in the same manner as amended claim 1. In addition, these claims recite additional limitations not shown or suggested by the cited combination. For example, claim 2 recites the undo and remaking operations recited in claim 1 and not taught or suggested by the cited combination.

Claims 5-11 and 13 have been rejected under 35 U.S.C. §103(a) as obvious over the Jain and LaRue references and further in view of U.S. Patent No. 5,802,322 (Niblett.) The examiner comments that Jain discloses most of the claimed material except that it does not disclose the use of endpoint numbers as identifiers. However, the examiner claims that Niblett shows such endpoint numbers and their use as identifiers. The examiner concludes that it would have been obvious to combine Jain, LaRue and Niblett in order to more completely serialize updates.

As mentioned in the response to a previous office communication, the Niblett patent discloses a token passing conference ring network in which a "permit-token" is used to serialize updates. When updates are performed, the permit-token is used to establish an order for the updates. In this manner, each node in the system will know when an update is missing. Consequently , the node can wait until the missing update has been received before applying later updates. This operation is clearly set forth in the Niblett abstract.

Claims 5-11 depend on amended claim 1 and incorporate the limitations thereof. The combination of Jain, LaRue and Niblett references does not suggest the make, undo and remake operations that are recited in amended claim 1. As discussed above, Jain and LaRue do not teach or suggest the recited combination. The addition of Niblett does not change this conclusion. This is because the Niblett serialization mechanism guarantees that an update cannot arrive "out-of-order" after other updates have been applied as in the present invention. Therefore, there is no reason to undo updates and then redo them in a different order as recited in amended claim 1 lines 17-22 as discussed above and Niblett does not mention any such operation. Consequently, the

combination of Jain, LaRue and Niblett cannot teach or suggest the recited limitations since no of the reference teaches or suggests the recited combination. Therefore, claims 5-11 patentably distinguishes over the cited combination of Jain, LaRue and Niblett.

Claim 13 is dependent on amended claim 12 and incorporates the limitations thereof. Since amended claim 12 recites essentially the same limitations as amended claim 1, claim 13 distinguishes over the cited combination in the same manner as amended claim 1.

New claims 35-52 have been added to more specifically recite the conditions under which the make, undo and redo operations are processed. These claims are based on Figures 11A, 11B and 12 and the accompanying description at page 33, line 14 to page 38, line 17. The two independent claims 35 and 44 have similar scope and claim 35 is representative of that scope. Claim 35 recites, "...making the data changes as specified by the first and second data change requests..." (lines 10-11); "when a change specified by one of the first and second data change requests cannot be made in the order determined by the data change requests because other changes have already been made to the local data copy, undoing the other changes to the local data copy..."(lines 13-16); making a change to the local data copy as specified by the one data change request; (lines 17-18) and "...redoing data changes undone ...so that data changes are made to the local data copy consistently with changes made to local data copies in all network-capable devices."(lines 19-21). Consequently, claims 35 and 44 recite the make, undo and redo operations that, as discussed above, are not taught or recited in Jain, LaRue and Niblett. Consequently, these claims patentably distinguish over the cited references.

Claims 36-43 are dependent, either directly or indirectly, on new claim 35 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination in the same manner as new claim 35. Similarly, claims 45-52 are dependent, either directly or indirectly, on new claim 44 and incorporate the limitations thereof. Therefore, they distinguish over the cited combination in the same manner as new claim 44.

In light of the forgoing amendments and remarks, this application is now believed in condition for allowance and a notice of allowance is earnestly solicited. If the examiner has any further questions regarding this amendment, he is invited to call applicants' attorney at the number listed below. The examiner is hereby authorized to charge any fees or direct any payment under 37 C.F.R. §§1.17, 1.16 to Deposit Account number 02-3038.

Respectfully submitted

Paul E. Kudirka Date: 11/26/03

Paul E. Kudirka, Esq. Reg. No. 26,931
KUDIRKA & JOBSE, LLP
Customer Number 021127
Tel: (617) 367-4600 Fax: (617) 367-4656